

Piazza (U.S. Patent No. 5,881,291). This rejection is hereby respectfully traversed.

Under 35 U.S.C. § 102, the Patent Office bears the burden of presenting at least a prima facie case of anticipation. In re Sun, 31 USPQ2d 1451, 1453 (Fed. Cir. 1993) (unpublished). Anticipation requires that a prior art reference disclose, either expressly or under the principles of inherency, each and every element of the claimed invention. Id. "In addition, the prior art reference must be enabling." Akzo N.V. v. U.S. International Trade Commission, 808 F.2d 1471, 1479, 1 USPQ2d 1241, 1245 (Fed. Cir. 1986), cert. denied, 482 U.S. 909 (1987). That is, the prior art reference must sufficiently describe the claimed invention so as to have placed the public in possession of it. In re Donohue, 766 F.2d 531, 533, 226 USPQ 619, 621 (Fed. Cir. 1985). "Such possession is effected if one of ordinary skill in the art could have combined the publication's description of the invention with his own knowledge to make the claimed invention." Id.

The Examiner asserts that "Piazza teaches a compiler and compilation method for processing a source program in a programming language in the Shceme/Lisp family into a

representation know as continuation-passing style (CPS) before generating object code, with optimization also being involved in the processing." Office Action, p. 2.

As stated in the specification, program-language compilers (such as the one disclosed in Piazza) are known. See specification p. 10, lines 6-9. However, Applicant is the first to apply compiling techniques to "C/C++ programs designed for a multitasking system into a new C/C++ program designed for a high performance run-to-completion system." Specification p. 10, lines 9-14. Applicant respectfully submits that each of the independent claims (i.e., 1, 6 and 11-13) are amended to reflect this novel feature and the Piazza fails to disclose or suggest compiling C/C++ programs. For at least this reason, Applicant respectfully requests that the rejections of claims 1, 6 and 11-12 be withdrawn.

Claims 2-5 and 7-10 are dependent upon one of independent claim 1 or claim 6. Thus, since independent claims 1 and 6 should be allowable as discussed above, claims 2-5 and 7-10 should also be allowable at least by virtue of their dependency on one of independent claims 1 or 6. Moreover, these claims recite additional features

which are not claimed, disclosed, or even suggested by the cited references taken either alone or in combination.

In view of the foregoing, it is respectfully requested that the aforementioned anticipation rejection of claims 1-12 be withdrawn.

II. THE OBVIOUSNESS REJECTION OF CLAIM 13

On page 5 of the Office Action, claim 13 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Piazza in view of Bak (U.S. Patent No. 6,704,927). This rejection is hereby respectfully traversed.

Under 35 U.S.C. § 103, the Patent Office bears the burden of establishing a prima facie case of obviousness. In re Fine, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988). The Patent Office can satisfy this burden only by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of references. Id. Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination. ACS Hospital Systems, Inc. v. Montefiore Hospital, 732 F.2d 1572, 1577,

221 USPQ 929, 933 (Fed. Cir. 1984). That is, under 35 U.S.C. § 103, teachings of references can be combined only if there is some suggestion or motivation to do so. Id. However, the motivation cannot come from the applicant's invention itself. In re Oetiker, 977 F.2d 1443, 1447, 24 USPQ2d 1443, 1446 (Fed. Cir. 1992). Rather, there must be some reason, suggestion, or motivation found in the prior art whereby a person of ordinary skill in the art would make the combination. Id.

As stated in MPEP § 2143, to establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). Also, as stated in MPEP § 2143.01, obviousness can only be

established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. In re Fine, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); In re Jones, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. In re Mills, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990). Further, as stated in MPEP § 2143.01, to establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). That is, "[a]ll words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson, 424 F.2d 1382, 165 USPQ 494, 496 (CCPA 1970). Additionally, as stated in MPEP § 2141.02, a prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention. W.L. Gore & Associates, Inc. v. Garlock, Inc., 721 F.2d 1540, 220 USPQ

303 (Fed. Cir. 1983), cert. denied, 469 U.S. 851 (1984). Finally, if an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. In re Fine, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

The Examiner asserts that Piazza discloses the invention recited in claim 13 except that Piazza "does not explicitly disclose a signal embodied in a carrier wave." Office Action, p. 5. Applicant respectfully submits that Piazza does not disclose transforming a C/C++ program as claimed in amended claim 13.

In view of the foregoing, it is respectfully requested that the aforementioned obviousness rejection of claim 13 be withdrawn.

III. CONCLUSION

In view of the foregoing, it is respectfully submitted that the present application is in condition for allowance, and an early indication of the same is courteously solicited. The Examiner is respectfully requested to contact the undersigned by telephone at the below listed telephone number, in order to expedite resolution of any issues and to expedite passage of the present application

to issue, if any comments, questions, or suggestions arise in connection with the present application.

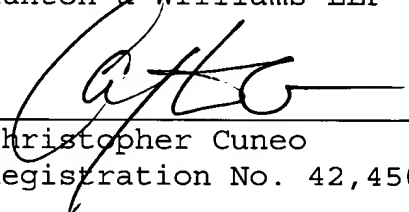
To the extent necessary, a petition for an extension of time under 37 CFR § 1.136 is hereby made.

Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 50-0206, and please credit any excess fees to the same deposit account.

Respectfully submitted,

Hunton & Williams LLP

By:


Christopher Cuneo
Registration No. 42,450

for
Thomas E. Anderson
Registration No. 37,063

CJC:dms

Hunton & Williams LLP
1900 K Street, N.W.
Washington, D.C. 20006-1109
Telephone: (202) 955-1500
Facsimile: (202) 778-2201

Date: June 28, 2004

APPENDIX A

1. (Currently Amended) A method for transforming a C/C++ program having a first multi-tasking property to a C/C++ program having a second multi-tasking property, the method comprising:

transforming a first C/C++ program having a first multi-tasking property into a data structure;
transforming the data structure to include an explicit multi-tasking transfer of control command;

optimizing the data structure to reduce an amount of program state that is saved at a transfer of control; and

generating a second C/C++ program having a second multi-tasking property using the optimized data structure

2. (Original) The method of claim 1, wherein the data structure further comprises a syntax tree.

3. (Original) The method of claim 2, wherein the step of transforming the data structure to include an explicit

multi-tasking transfer of control command further
comprises:

converting the syntax tree to a continuation-
passing style (CPS).

4. (Original) The method of claim 1, wherein the first multi-tasking property comprises a property relating to a preemptive multitasking model and the second multi-tasking property comprises a property relating to a run-to-completion model.
5. (Original) The method of claim 1, wherein the first program having a first multi-tasking property operates using a first program language and the second program having a second multi-tasking property also operates using the first program language.
6. (Currently Amended) A system for transforming a C/C++ program having a first multi-tasking property to a C/C++ program having a second multi-tasking property, the system comprising:

a data structure transformer for transforming a first C/C++ program having a first multi-tasking property into a data structure;

a multi-tasking transformer for transforming the
data structure to include an explicit multi-
tasking transfer of control command;
a program state optimizer for optimizing the data
structure to reduce an amount of program
state that is saved at a transfer of
control; and
a program generator for generating a second C/C++
program having a second multi-tasking
property using the optimized data structure.

7. (Original) The system of claim 6, wherein the data
structure further comprises a syntax tree.

8 (Original). The system of claim 7, wherein the multi-
tasking transformer further comprises:
a converter for converting the syntax tree to a
continuation-passing style (CPS).

9. (Original) The system of claim 6, wherein the first
multi-tasking property comprises a property relating
to a preemptive multitasking model and the second
multi-tasking property comprises a property relating
to a run-to-completion model.

- 10 (Original). The system of claim 6, wherein the first program having a first multi-tasking property operates using a first program language and the second program having a second multi-tasking property also operates using the first program language.
- 11 (Currently Amended). An article of manufacture for transforming a C/C++ program having a first multi-tasking property to a C/C++ program having a second multi-tasking property, the article of manufacture comprising:
- at least one processor readable carrier; and
 - instructions carried on the at least one carrier; wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:
 - transform a first C/C++ program having a first multi-tasking property into a data structure;
 - transform the data structure to include an explicit multi-tasking transfer of control command;

optimize the data structure to reduce an
amount of program state that is saved
at a transfer of control; and
generate a second C/C++ program having a
second multi-tasking property using the
optimized data structure.

12. (Currently Amended) A processor readable medium for
providing instructions to at least one processor for
directing the at least one processor to:
transform a first C/C++ program having a first multi-
tasking property into a data structure;

transform the data structure to include an
explicit multi-tasking transfer of control
command;
optimize the data structure to reduce an amount
of program state that is saved at a transfer
of control; and
generate a second C/C++ program having a second
multi-tasking property using the optimized
data structure.

13. (Currently Amended) A signal embodied in a carrier wave and
representing sequences of instructions which, when executed

by at least one processor, cause the at least one processor to transform a program having a first multi-tasking property to a program having a second multi-tasking property by performing the steps of:

transforming a first C/C++ program having a first multi-tasking property into a data structure;

transforming the data structure to include an explicit multi-tasking transfer of control command;

optimizing the data structure to reduce an amount of program state that is saved at a transfer of control; and

generating a second C/C++ program having a second multi-tasking property using the optimized data structure.